

FIG. 1

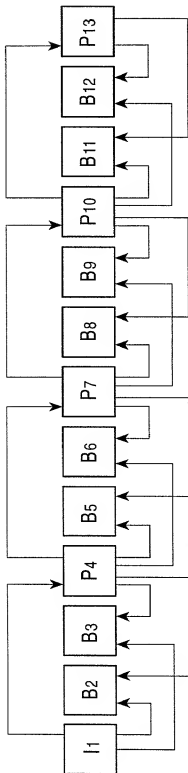


FIG. 2

Profile and Level	UPPER LIMIT				
	Bit rates (Mbit/s)	Samples/line	Lines/Frame	Frames/sec	Samples/sec
4:2:2P@HL	300	1920	1152	60	62,668,800
4:2:2P@ML	50	720	608	30	11,059,200
MP@HL	80	1920	1152	60	62,668,800
MP@HL-1440	60	1440	1152	60	47,001,600
MP@ML	15	720	576	30	10,368,000
MP@LL	4	352	288	30	3,041,280
SP@ML	15	720	576	30	10,368,000

FIG. 4A

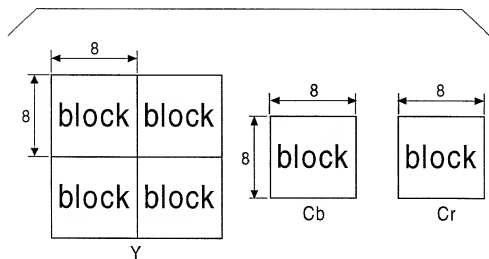


FIG. 4B

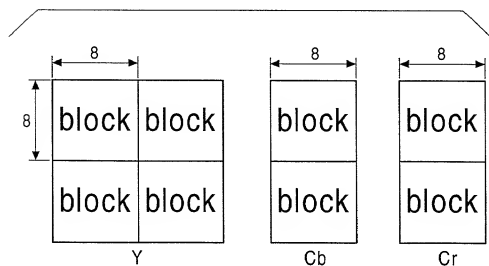


FIG. 5

sequence_header () {	NUMBER OF BITS	MNEMONIC
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantiser_matrix	1	
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	uimsbf
next_start_code ()		
}		

FIG. 6

sequence_extension () {	NUMBER OF BITS	MNEMONIC
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code ()		
}		

FIG. 7

group_of_picture_header () {	NUMBER OF BITS	MNEMONIC
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code ()		
}		

FIG. 8

picture_header () {	NUMBER OF BITS	MNEMONIC
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 picture_coding_type==3){		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if(picture_coding_type==3)		
full_pel_forward_vector	1	
backward_f_code	3	uimsbf
}		
while(nextbits()=="1") {		
extra_bit_picture/*with the value "1"*/	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture/*with the value "0"*/	1	uimsbf
next_start_code ()t		
}		

FIG. 9

picture_coding_extension () {	NUMBER OF BITS	MNEMONIC
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code[0][0]/*forward horizontal*/	4	uimsbf
f_code[0][1]/*forward vertical*/	4	uimsbf
f_code[1][0]/*backward horizontal*/	4	uimsbf
f_code[1][1]/*backward vertical*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag) {		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code ()		
}		

FIG. 10

picture_data () {	NUMBER OF BITS	MNEMONIC
do {		
slice ()		
} while (nextbits()==slice_start_code)		
next_start_code()		
}		

09866288.052501

FIG. 11

slice () {	NUMBER OF BITS	MNEMONIC
slice_start_code	32	bslbf
if(vertical_size>2800)		
slice_vertical_position_extention	3	uimsbf
if(<sequence_scalable_extention() is present in the bitstream>)		
if(<scalable_mode=="data partitioning">)		
priority_breakpoint	7	uimsbf
quantiser_scale_code	5	uimsbf
if(nextbits()=="1" {		
intra_slice_flag	1	uimsbf
intra_slice_	1	uimsbf
reserved_bits	7	uimsbf
while(nextbits()=="1" {		
extra_bit_slice/*with the value "1"*/	1	uimsbf
extra_information_slice	8	uimsbf
}		
}		
extra_bit_slice/*with the value "0"*/	1	uimsbf
do {		
macroblock ()		
} while (nextbits()!="000 0000 0000 0000 0000 0000")		
next_start_code () {		
}		

FIG. 12

macroblock () {	NUMBER OF BITS	MNEMONIC
while (nextbits()=='0000 0001 000')		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	vclbf
macroblock_modes()		
if(macroblock_quant)		
quantiser_scale_code	5	uimsbf
if(macroblock_motion_forward (macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if(macroblock_motion_backward)		
motion_vectors(1)		
if(macroblock_intra && concealment_motion_vectors))		
marker_bit	1	bslbf
if(macroblock_pattern)		
coded_block_pattern()		
for(1=0; j<block_count; i++) {		
block(i)		
}		
}		

FIG. 13

macroblock_modes () {	NUMBER OF BITS	MNEMONIC
macroblock_type	1-9	vlc1bf
if((spatial_temporal_weight_code_flag==1)&& (spatial_temporal_weight_code_table_index!="00")) {		
spatial_temporal_weight_code	2	uimsbf
}		
if(macroblock_motion_forward macroblock_motion_backward) {		
if(picture_structure=="frame") {		
if(frame_pred_frame_dct==0)		
frame_motion_type	2	uimsbf
} else {		
field_motion_type	2	uimsbf
}		
}		
if((picture_structure=="frame picture") && (frame_pred_frame_dct==0) && (macroblock_intra1macroblock_pattern)) {		
dct_type	1	uimsbf
}		
}		

FIG. 14

NAME	Start code value
Picture_start_code	00
Slice_start_code	01 ~ AF
Reserved	B0
Reserved	B1
User_data_start_code	B2
Sequence_header_code	B3
Sequence_error_code	B4
Extension_start_code	B5
Reserved	B6
Sequence_end_code	B7
Group_start_code	B8
System_start_code	B9 ~ FF

FIG. 15

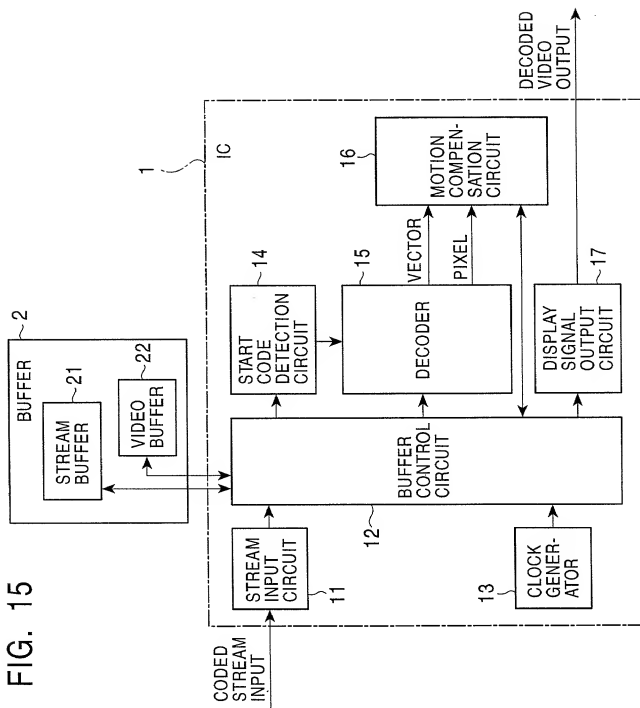


FIG. 16

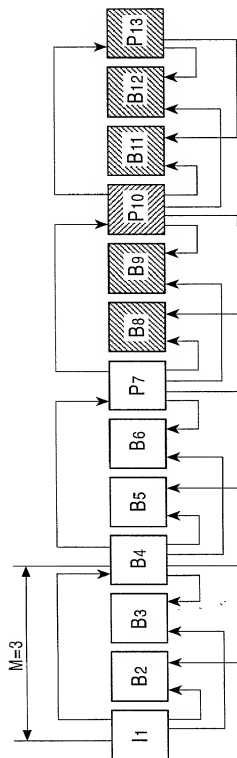
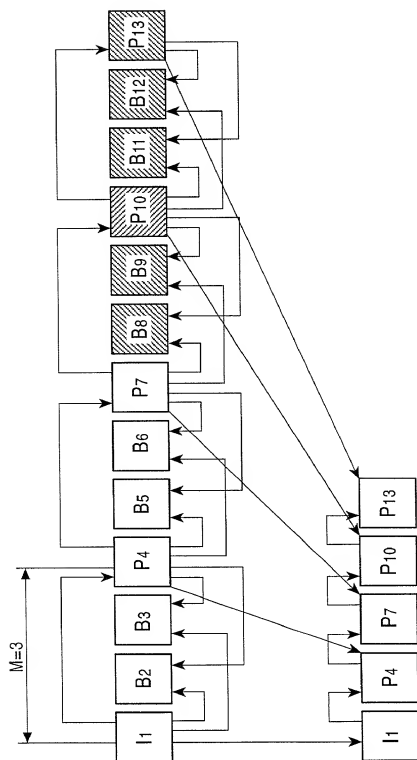


FIG. 17



101 102 103 104

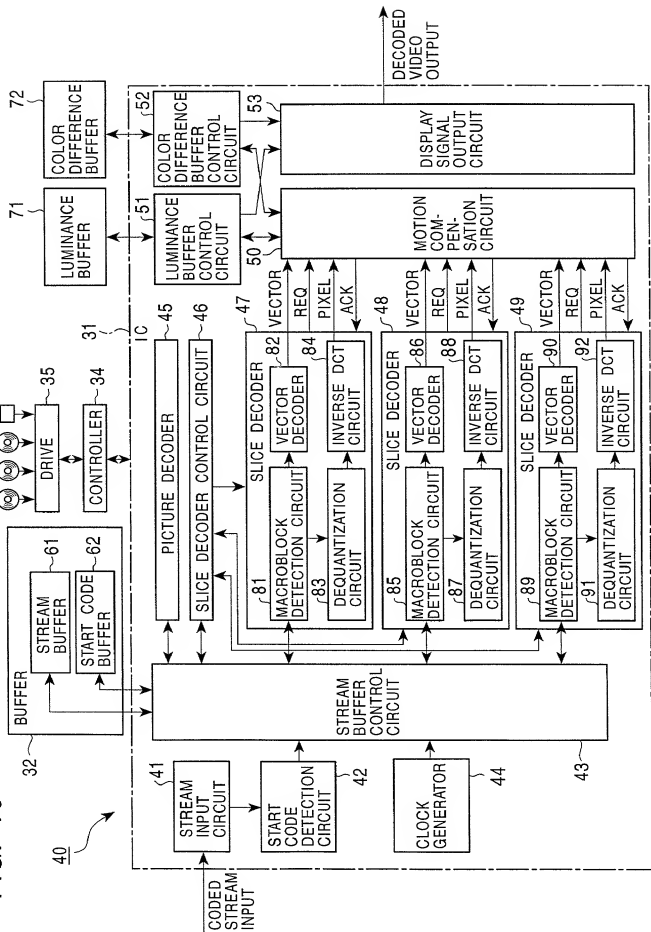


FIG. 19

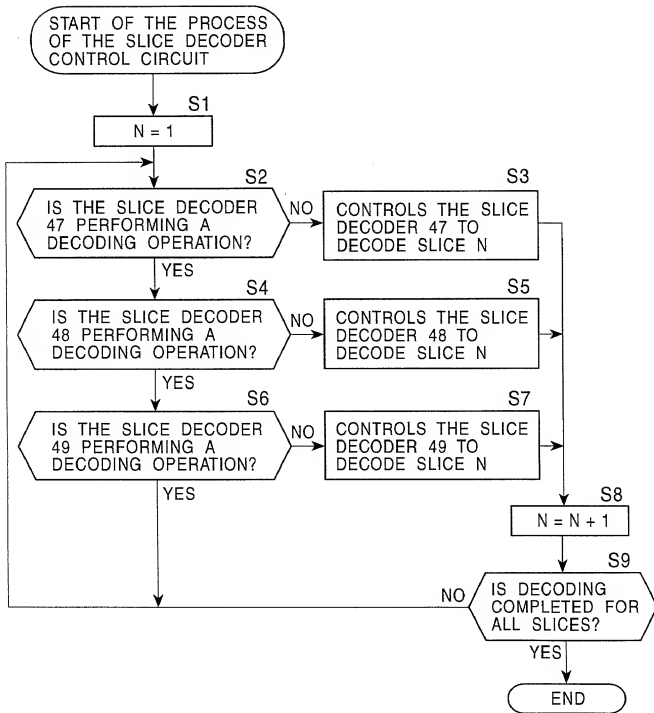


FIG. 20

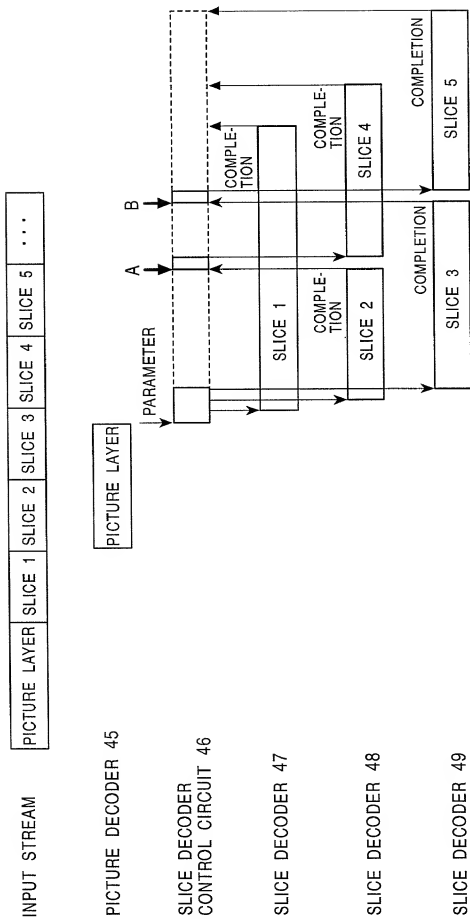


FIG. 21

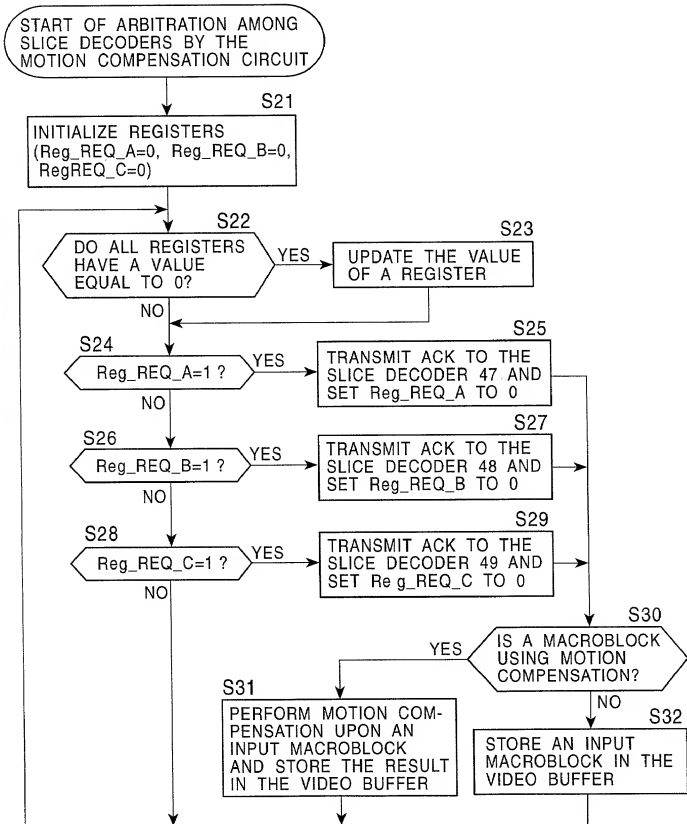


FIG. 22

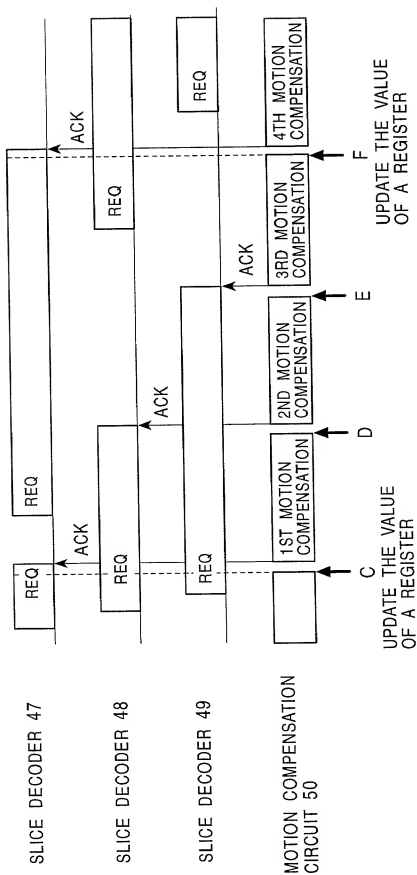


FIG. 23

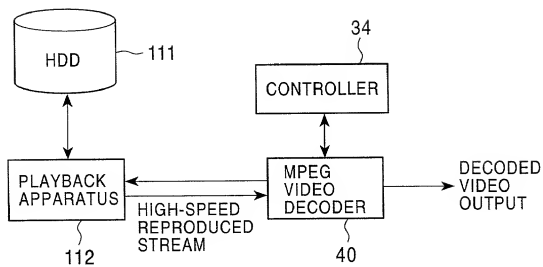


FIG. 24

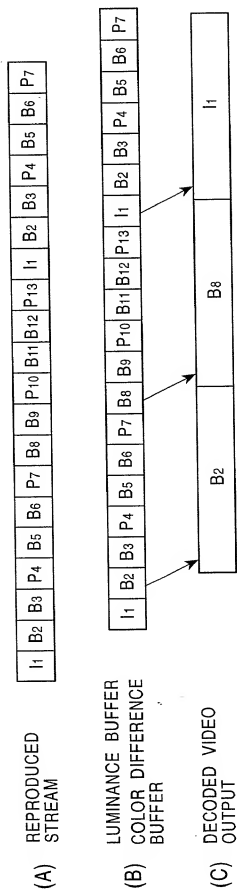


FIG. 25

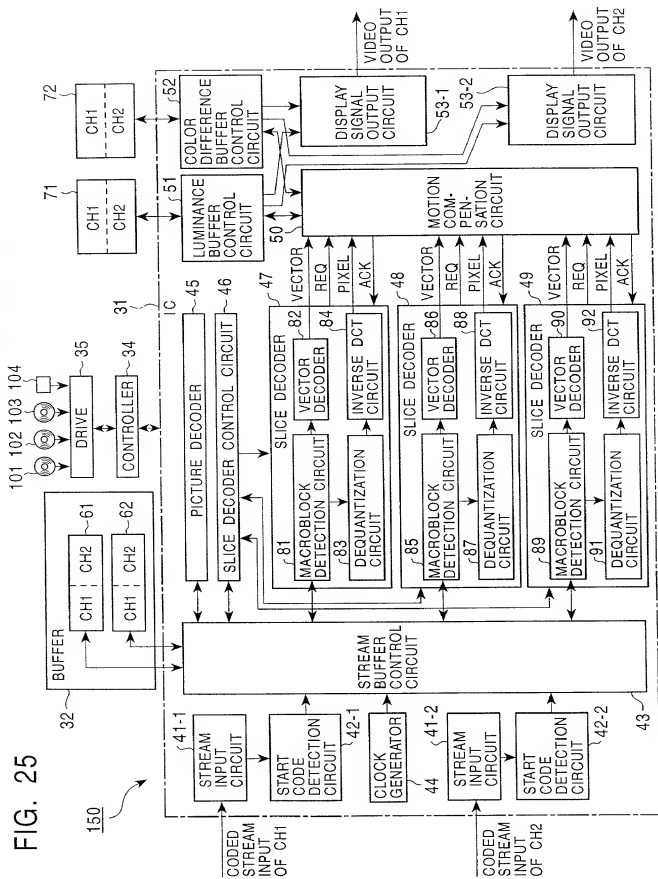


FIG. 26

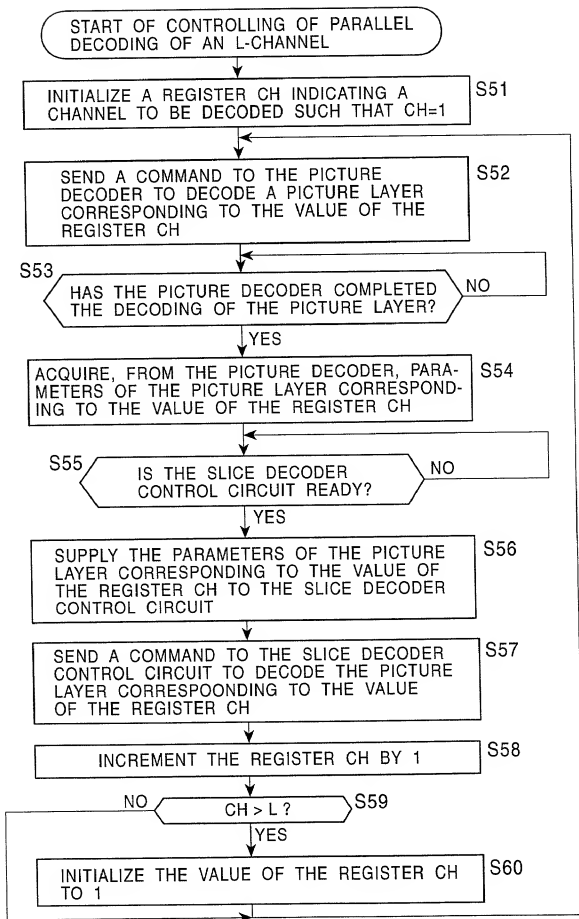


FIG. 27

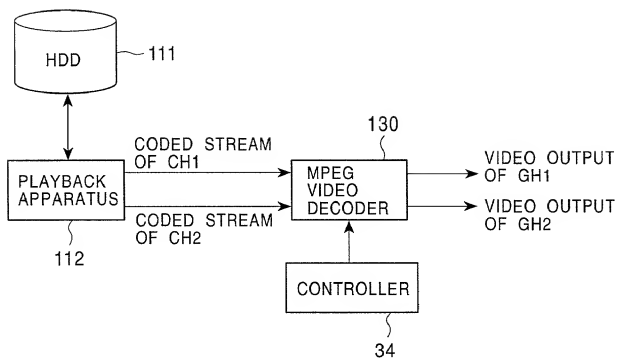


FIG. 28

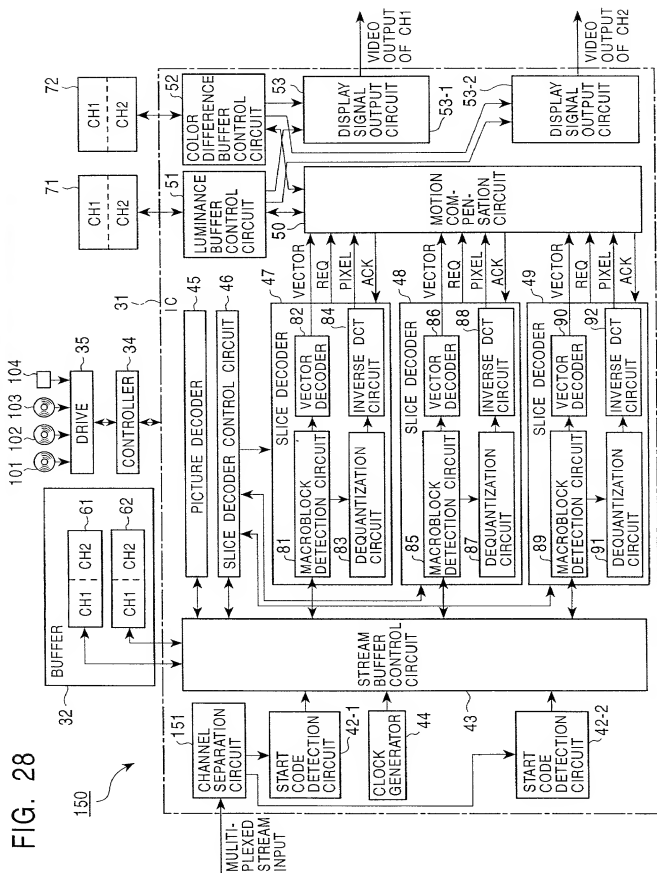


FIG. 29

